

EEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTT
EEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTT
EEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTT
EEE	DDD	TTT
EEEEEEEEE	DDD	TTT
EEEEEEEEE	DDD	TTT
EEEEEEEEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEE	DDDDDDDDDDDDDD	TTT

FILEID**CHMGBUF

H 9

CCCCCCCC HH HH MM MM MM GGGGGGGG BBBB BBBB UU UU FFFFFFFF
CCCCCCCC HH HH MM MM MM GGGGGGGG BBBB BBBB UU UU FFFFFFFF
CC HH HH MMMM MMMM GG BB BB UU UU FF
CC HH HH MMMM MMMM GG BB BB UU UU FF
CC HH HH MM MM MM GG BB BB UU UU FF
CC HH HH MM MM MM GG BBBB BBBB UU UU FF
CC HHHHHHHHHHHH MM MM GG BBBB BBBB UU UU FFFFFFFF
CC HHHHHHHHHHHH MM MM GG BBBB BBBB UU UU FFFFFFFF
CC HH HH MM MM GG GGGGGG BB BB UU UU FF
CC HH HH MM MM GG GGGGGG BB BB UU UU FF
CC HH HH MM MM GG GG BB BB UU UU FF
CC HH HH MM MM GG GG BB BB UU UU FF
CCCCCCCC HH HH MM MM GGGGGG BBBB BBBB UUUUUUUUUU FF
CCCCCCCC HH HH MM MM GGGGGG BBBB BBBB UUUUUUUUUU FF

....
....
....

LL IIIII SSSSSSS
LL IIIII SSSSSSS
LL II SS
LL II SS
LL II SS
LL II SSSSS
LL II SSSSS
LL II SS
LL II SS
LL II SS
LLLLLLLLL LIII SSSSSSS
LLLLLLLLL LIII SSSSSSS

EF
VC

```
0001 0 %TITLE 'EDTSCHMGBUF - look for a buffer name'  
0002 0 MODULE EDTSCHMGBUF (  
0003 0 IDENT = 'V04-000'  
0004 0 ) =  
0005 1 BEGIN  
0006 1 *****  
0007 1 *  
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0010 1 * ALL RIGHTS RESERVED.  
0011 1 *  
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0017 1 * TRANSFERRED.  
0018 1 *  
0019 1 *  
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0022 1 * CORPORATION.  
0023 1 *  
0024 1 *  
0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0027 1 *  
0028 1 *****  
0029 1  
0030 1  
0031 1 **  
0032 1 FACILITY: EDT -- The DEC Standard Editor  
0033 1  
0034 1 ABSTRACT:  
0035 1  
0036 1 This module scans the command buffer for a buffer name.  
0037 1  
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant  
0039 1  
0040 1 AUTHOR: Bob Kuslits, CREATION DATE: Unknown  
0041 1  
0042 1 MODIFIED BY:  
0043 1  
0044 1 1-001 - Original. DJS 04-Feb-1981. This module was created by  
0045 1 extracting the routine EDT$GET_BUF from the module CHANGE.BLI.  
0046 1 1-002 - Regularize headers. JBS 02-Mar-1981  
0047 1 1-003 - Use new message codes. JBS 04-Aug-1981  
0048 1 1-004 - Check first to see if there are more characters on command line.  
0049 1 STS 21-Jun-1982  
0050 1 1-005 - Change alphabetic test. JBS 19-Jul-1982  
0051 1 1-006 - Add conditional for VT220 support. JBS 11-Feb-1983  
0052 1 --  
0053 1
```

```
: 55      0054 1 %SBTTL 'Declarations'  
: 56      0055 1  
: 57      0056 1 TABLE OF CONTENTS:  
: 58      0057 1  
: 59      0058 1  
: 60      0059 1 REQUIRE 'EDTSRC:TRAROUNAM';  
: 61      0498 1  
: 62      0499 1 FORWARD ROUTINE  
: 63      0500 1     EDT$GET_BUF : NOVALUE; ! Look for a buffer specification for the cut, paste, and append commands  
: 64      0501 1  
: 65      0502 1  
: 66      0503 1 INCLUDE FILES:  
: 67      0504 1 !  
: 68      0505 1  
: 69      0506 1 REQUIRE 'EDTSRC:EDTREQ';  
: 70      0641 1  
: 71      0642 1 LIBRARY 'EDTSRC:SUPPORTS';  
: 72      0643 1  
: 73      0644 1  
: 74      0645 1 MACROS:  
: 75      0646 1  
: 76      0647 1     NONE  
: 77      0648 1  
: 78      0649 1 EQUATED SYMBOLS:  
: 79      0650 1  
: 80      0651 1     NONE  
: 81      0652 1  
: 82      0653 1 OWN STORAGE:  
: 83      0654 1  
: 84      0655 1     NONE  
: 85      0656 1  
: 86      0657 1 EXTERNAL REFERENCES:  
: 87      0658 1  
: 88      0659 1     In the routine
```

```
90      0660 1 %SBTTL 'EDT$GET_BUF - look for a buffer name'
91      0661 1
92      0662 1 GLOBAL ROUTINE EDT$GET_BUF           ! Look for a buffer name
93          : NOVALUE =
94
95      0665 1 ++
96      0666 1 | FUNCTIONAL DESCRIPTION:
97          0667 1
98          0668 1 | This routine looks for a buffer spec for use by cut/paste and append.
99          0669 1 | If an '=' is seen, an explicit buffer spec is there; otherwise, use
100         0670 1 | the paste buffer. The routine places the address of the buffer in
101         0671 1 | EDT$SA_ALT_BUF, with a zero indicating failure.
102         0672 1
103         0673 1 FORMAL PARAMETERS:
104         0674 1
105         0675 1     NONE
106         0676 1
107         0677 1 IMPLICIT INPUTS:
108         0678 1
109         0679 1     EDT$SA_CMD-END
110         0680 1     EDT$SA_CMD-BUF
111         0681 1     EDT$SA_PST-BUF
112         0682 1     EDT$SA_CUR-BUF
113         0683 1
114         0684 1 IMPLICIT OUTPUTS:
115         0685 1
116         0686 1     EDT$SA_ALT-BUF
117         0687 1     EDT$SA_CMD-BUF
118         0688 1
119         0689 1 ROUTINE VALUE:
120         0690 1
121         0691 1     1 = buffer spec seen, 0 = no buffer spec seen
122         0692 1
123         0693 1 SIDE EFFECTS:
124         0694 1
125         0695 1     Uses EDT$FND_BUF to create the buffer if necessary. If it fails,
126         0696 1     gives an error message.
127         0697 1
128         0698 1 --+
129         0699 1
130         0700 2 BEGIN
131         0701 2
132         0702 2 EXTERNAL ROUTINE
133             0703 2     EDT$MSG_BELL : NOVALUE,           | Output a message to the terminal with a warning bell
134             0704 2     EDT$SCNV_UPC,                 | Convert string to upper case
135             0705 2     EDT$FND_BUF,                  | Search for a text buffer
136             0706 2     EDT$SRD_CURLN;                | Get the current line
137             0707 2
138             0708 2 EXTERNAL
139             0709 2     EDT$SA_CMD-END,            | End of command pointer
140             0710 2     EDT$SA_CMD-BUF,            | Command string pointer
141             0711 2     EDT$SA_ALT-BUF : REF TBCB_BLOCK, | Alternate buffer used for cut/paste.
142             0712 2     EDT$SA_PST-BUF : REF TBCB_BLOCK, | The paste buffer tpcb
143             0713 2     EDT$SA_CUR-BUF : REF TBCB_BLOCK, | The current buffer tccb
144             0714 2
145             L 0715 2 %IF SUPPORT_VT220
146             0716 2 %THEN
```

```
147      0717 2
148      0718 2
149      0719 2  XFI   EDT$SB_CHAR_INFO : BLOCKVECTOR [256, 1, BYTE] ! Information about characters
150      0720 2
151      0721 2
152      0722 2
153      0723 2
154      0724 2
155      0725 2
156      0726 2
157      0727 2
158      0728 2
159      0729 2
160      0730 2
161      0731 2
162      0732 2
163      0733 2
164      0734 2
165      0735 3
166      0736 2  THEN
167      0737 3
168      0738 3
169      0739 3
170      0740 3
171      0741 3
172      0742 4
173      0743 4
174      0744 4
175      L 0745 4  XIF SUPPORT_VT220
176      0746 4  XTHEN
177      0747 4
178      0748 6
179      0749 6
180      0750 6
181      0751 5
182      0752 4
183      0753 4
184      0754 4
185      U 0755 4  XELSE
186      U 0756 4
187      U 0757 4
188      U 0758 4
189      U 0759 4
190      U 0760 4
191      U 0761 4
192      U 0762 4
193      U 0763 4
194      0764 4  XFI
195      0765 4
196      0766 4
197      0767 4
198      0768 3
199      0769 3
200      0770 3
201      0771 3
202      L 0772 3  XIF SUPPORT_VT220
203      0773 3  XTHEN

      0718 2
      0719 2
      0720 2
      0721 2
      0722 2
      0723 2
      0724 2
      0725 2
      0726 2
      0727 2
      0728 2
      0729 2
      0730 2
      0731 2
      0732 2
      0733 2
      0734 2
      0735 3
      0736 2
      0737 3
      0738 3
      0739 3
      0740 3
      0741 3
      0742 4
      0743 4
      0744 4
      0745 4
      0746 4
      0747 4
      0748 6
      0749 6
      0750 6
      0751 5
      0752 4
      0753 4
      0754 4
      U 0755 4
      U 0756 4
      U 0757 4
      U 0758 4
      U 0759 4
      U 0760 4
      U 0761 4
      U 0762 4
      U 0763 4
      0764 4
      0765 4
      0766 4
      0767 4
      0768 3
      0769 3
      0770 3
      0771 3
      L 0772 3
      0773 3

      NAMELEN = .NAMELEN + 1;
      EDTSSA_CMD_BUF = CH$PLUS (.EDTSSA_CMD_BUF, 1);
      END;

      CHAR = CH$RCHAR (.NAME_START);
```

EDT\$CHMGBUF
V04-000

EDT\$CHMGBUF - Look for a buffer name
EDT\$GET_BUF - look for a buffer name

M 9
15-Sep-1984 23:57:05 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 12:22:31 DISK\$VMSMASTER:[EDT.SRC]CHMGBUF.BLI;1 Page 5
(3)

ED
VC

```
: 204      0774 3
: 205      0775 4      IF ((.NAME_LEN EQL 0) OR (.EDT$SB_CHAR_INFO [.CHAR, 0, 0, 2, 0] EQL 0))
: 206      0776 3      THEN
: 207      U 0777 3      %ELSE
: 208      U 0778 3
: 209      U 0779 3      IF ((.NAME_LEN EQL 0) OR
: 210      U 0780 3          (.CHAR_LSS XC'A') OR ((.CHAR GTR XC'Z') AND (.CHAR LSS XC'a')) OR (.CHAR GTR XC'z'))
: 211      U 0781 3      THEN
: 212      U 0782 3      %FI
: 213      U 0783 3
: 214      U 0784 4      BEGIN
: 215      U 0785 4          EDT$MSG_BELL (EDTS_INBUFNAM);
: 216      U 0786 4          RETURN;
: 217      U 0787 3          END;
: 218      U 0788 3
: 219      U 0789 3      EDT$CNV UPC (.NAME_START, .NAME_LEN);
: 220      U 0790 3      SAVE_TBCB = .EDT$SA_CUR_BUF;
: 221      U 0791 3
: 222      U 0792 3      IF EDT$FND_BUF (.NAME_START, .NAME_LEN)
: 223      U 0793 3      THEN
: 224      U 0794 3          EDT$SA_ALT_BUF = .EDT$SA_CUR_BUF
: 225      U 0795 3      ELSE
: 226      U 0796 3          EDT$MSG_BELL (EDTS_INSMEM);
: 227      U 0797 3
: 228      U 0798 3      EDT$SA_CUR_BUF = .SAVE_TBCB;
: 229      U 0799 3      EDT$RD_CURLN ();
: 230      U 0800 3      END
: 231      U 0801 2      ELSE
: 232      U 0802 2          EDT$SA_ALT_BUF = .EDT$SA_PST_BUF;
: 233      U 0803 2
: 234      U 0804 1      END;
:                      ! of routine EDT$GET_BUF
```

.TITLE EDT\$CHMGBUF EDT\$CHMGBUF - look for a buffer nam
e
.IDENT \V04-000\

.EXTRN EDT\$MSG_BELL, EDT\$CNV UPC
.EXTRN EDT\$FND_BUF, EDT\$RD_CURLN
.EXTRN EDT\$SA_CMD_END, EDT\$SA_CMD_BUF
.EXTRN EDT\$SA_ALT_BUF, EDT\$SA_PST_BUF
.EXTRN EDT\$SA_CUR_BUF, EDT\$SB_CHAR_INFO
.EXTRN EDTS_INBUFNAM, EDTS_INSMEM

.PSECT _EDT\$CODE,NOWRT, SHR, PIC,2

07FC 00000

5A 00000000G	00 9E 00002
59 00000000G	00 9E 00009
58 00000000G	00 9E 00010
57 00000000G	00 9E 00017
56 00000000G	00 9E 0001E
55 00000000G	00 9E 00025
	54 D4 0002C
51	65 D0 0002E
50	61 9A 00031

.ENTRY EDT\$GET_BUF, Save R2,R3,R4,R5,R6,R7,R8,R9,-: 0662
R10
MOVAB EDT\$SA_ALT_BUF, R10
MOVAB EDT\$MSG_BELL, R9
MOVAB EDT\$SB_CHAR_INFO, R8
MOVAB EDT\$SA_CMD_END, R7
MOVAB EDT\$SA_CUR_BUF, R6
MOVAB EDT\$SA_CMD_BUF, R5
CLRL NAMELEN
MOVL EDT\$SA_CMD_BUF, R1
MOVZBL (R1), CHAR

0732
0733

EDTSCHMGBUF
V04-000

EDTSCHMGBUF - look for a buffer name
EDT\$GET_BUF - look for a buffer name

N 9
15-Sep-1984 23:57:05
14-Sep-1984 12:22:31
VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDIT.SRC]CHMGBUF.BLI;1

Page 6
(3)

ED
VO

3D	50	D1	00034	CMP.L	CHAR, #61	0735
	78	12	00037	BNEQ	8\$:
67	51	D1	00039	CMP.L	R1, EDT\$SA_CMD_END	:
	76	1E	0003C	BGEQU	8\$:
52	65	D6	0003E	INCL	EDT\$SA_CMD_BUF	0738
51	65	D0	00040	MOVL	EDT\$SA_CMD_BUF, NAME_START	0739
67	65	D0	00043	1\$: MOVL	EDT\$SA_CMD_BUF, R1	0741
	51	D1	00046	CMP.L	R1, EDT\$SA_CMD_END	:
	21	1E	00049	BGEQU	3\$:
51	50	61	9A	0004B	MOVZBL	0743
	50	58	C1	0004E	(R1), CHAR	0744
03		61	93	00052	ADDL3	0745
		0F	12	00055	BITB	0746
F0	8F	61	91	00057	BNEQ	0747
		09	13	00058	BEQL	0748
0000005F	8F	50	D1	0005D	CMP.L	0749
		06	12	00064	BNEQ	0750
		54	D6	00066	2\$: INCL	0751
		65	D6	00068	NAME LEN	0752
		07	11	0006A	INCL	0753
50		62	9A	0006C	MOVZBL	0754
		54	D5	0006F	(NAME START), CHAR	0755
		06	13	00071	TSTL	0756
03		6840	93	00073	BEQL	0757
		0A	12	00077	BITB	0758
	00000000G	8F	DD	00079	5\$: EDT\$SB_CHAR_INFO[CHAR], #3	0759
69		01	FB	0007F	PUSHL	0760
		04	00082	CALLS	#EDTS_INBUFNAM	0761
				RET	#1, EDT\$MSG_BELL	0762
00000000G	00	14	BB	00083	PUSHR	0763
53		02	FB	00085	CALLS	0764
		66	D0	0008C	EDT\$SA_CUR_BUF, SAVE_TBCB	0765
00000000G	00	14	BB	0008F	PUSHR	0766
05		02	FB	00091	CALLS	0767
6A		50	E9	00098	BLBC	0768
		66	D0	0009B	MOVL	0769
		09	11	0009E	EDT\$SA_CUR_BUF, EDT\$SA_ALT_BUF	0770
00000000G	8F	DD	000A0	6\$: BRB	0771	
69		01	FB	000A6	PUSHL	0772
00000000G	66	53	D0	000A9	CALLS	0773
00		00	FB	000AC	EDT\$INSMEM	0774
		04	000B3	RET	#1, EDT\$MSG_BELL	0775
6A 00000000G	00	D0	000B4	8\$: MOVL	SAVE_TBCB, EDT\$SA_CUR_BUF	0776
		04	000B8	RET	#0, EDT\$RD_CURLN	0777
					EDT\$SA_PST_BUF, EDT\$SA_ALT_BUF	0778
					0779	0798
					0780	0799
					0781	0799
					0782	0799
					0783	0799
					0784	0799
					0785	0799
					0786	0799
					0787	0799
					0788	0799
					0789	0799
					0790	0799
					0791	0799
					0792	0799
					0793	0799
					0794	0799
					0795	0799
					0796	0799
					0797	0799
					0798	0799
					0799	0799
					0800	0799
					0801	0799
					0802	0799
					0803	0799
					0804	0799

: Routine Size: 188 bytes. Routine Base: _EDT\$CODE + 0000

: 235 0805 1
: 236 0806 1 !<BLF/PAGE>

EDTSCHMGBUF
V04-000

EDTSCHMGBUF - look for a buffer name
EDT\$GET_BUF - look for a buffer name

B 10
15-Sep-1984 23:57:05 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:22:31 DISK\$VMSMASTER:[EDT.SRC]CHMGBUF.BLI;1 Page 7
(4)

: 279 0807 1 END
: 39 0808 1
: 240 0809 0 ELUDOM

: ! of module EDTSCHMGBUF

E
VI

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	188	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
-\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	26	6	40	00:00.2
-\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1
-\$255\$DUA28:[EDT.SRC]SUPPORTS.L32;1	2	1	50	5	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:CHMGBUF/OBJ=OBJ\$:CHMGBUF MSRC\$:\$CHMGBUF.BLI/UPDATE=(ENH\$:\$CHMGBUF)

Size: 188 code + 0 data bytes
Run Time: 00:14.4
Elapsed Time: 00:17.7
Lines/CPU Min: 3370
Lexemes/CPU-Min: 11166
Memory Used: 95 pages
Compilation Complete

0131 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CHMFINENT LIS	CHMGCOUNT LIS	CHMGINSTR LIS	CHMGSUSTR LIS	CHMINIT LIS	CHMINSMOD LIS
CHMEMESS LIS	CHMGINSTR LIS	CHMGINSTR LIS	CHMGINSTR LIS	CHMINSTAB LIS	CHMINSTAB LIS
CHMENTTRM LIS	CHMGBUF LIS	CHMGINSTR LIS	CHMGINSTR LIS	CHMINSTR LIS	CHMINSTR LIS
CHMEXVERB LIS	CHMGOIR LIS	CHMGOIR LIS	CHMHLPKPD LIS	CHMINDATE LIS	CHMINDATE LIS
CHMENDWRD LIS	CHMGOISTR LIS	CHMGOISTR LIS	CHMINSCHR LIS	CHMINSCHR LIS	CHMINSCHR LIS
CHMEXCOM LIS	CHMKEYRD LIS	CHMKEYRD LIS	CHMKEYRD LIS	CHMKEYRD LIS	CHMKEYRD LIS